Reinforcement Learning Assignment 01

Waruni Alagalle Hitgedara (202487136)
Prasanth Senthan (202398190)
Memorial University of Newfoundland, Canada

Spring 2025

Contents

Module information

This document fulfills the requirements for the Reinforcement Learning Module (DSCI-6650-001).

1 Part 01: Stationary bandit problem

1.1 Environment setup

A 10-armed bandit environment is considered, where each arm's reward is drawn from a normal distribution. The true action values (means) μ_i for arms $i = 1, \ldots, 10$ are independently sampled from a standard normal distribution $\mathcal{N}(0, 1)$. The reward for each arm at each step is sampled from $\mathcal{N}(\mu_i, 1)$.

1.2 Algorithms implemented

- Greedy algorithm
- Epsilon-Greedy algorithm
- Optimistic Greedy algorithm
- Gradient Bandit algorithm

1.3 Simulation setup

Each algorithm was executed over 1,000 independent simulations, with each simulation consisting of 2,000 time steps. The performance of the algorithms was evaluated based on the following criteria:

- Average reward per time step
- Percentage of optimal actions selected over time

1.4 Reproducibility measures

To ensure the reproducibility of results across multiple runs, specific steps were taken to control randomness and maintain a consistent experimental environment:

- **Random seed initialization:** A fixed random seed was set for each simulation iteration using `np.random.seed()` to ensure that the generated action values and rewards remain consistent across runs.
- **Consistent simulation parameters:** All algorithms were run with the same number of arms, time steps, and simulations to enable fair and repeatable comparisons.

- **Fixed environment structure:** The true means for each arm were drawn from the same distribution ($\mathcal{N}(0, 1)$) in each simulation, controlled by the fixed seed.
- **Library consistency:** The implementation was run in a controlled Python environment with consistent versions of NumPy and Matplotlib to avoid discrepancies due to version differences.

1.5 Results and analysis

1.5.1 Greedy with non-optimistic initial values

The Greedy algorithm is applied to a 10-armed bandit problem using non-optimistic initial values. For each of the 1,000 simulations, the true reward values of the arms are drawn from a normal distribution. At each time step, the algorithm selects the arm with the highest estimated reward. In cases where multiple arms share the same highest estimate, one is selected at random. The estimated rewards are updated incrementally using the sample average of observed rewards. While the Greedy strategy quickly exploits known good actions, its lack of exploration may cause it to miss better options that have not yet been tried.

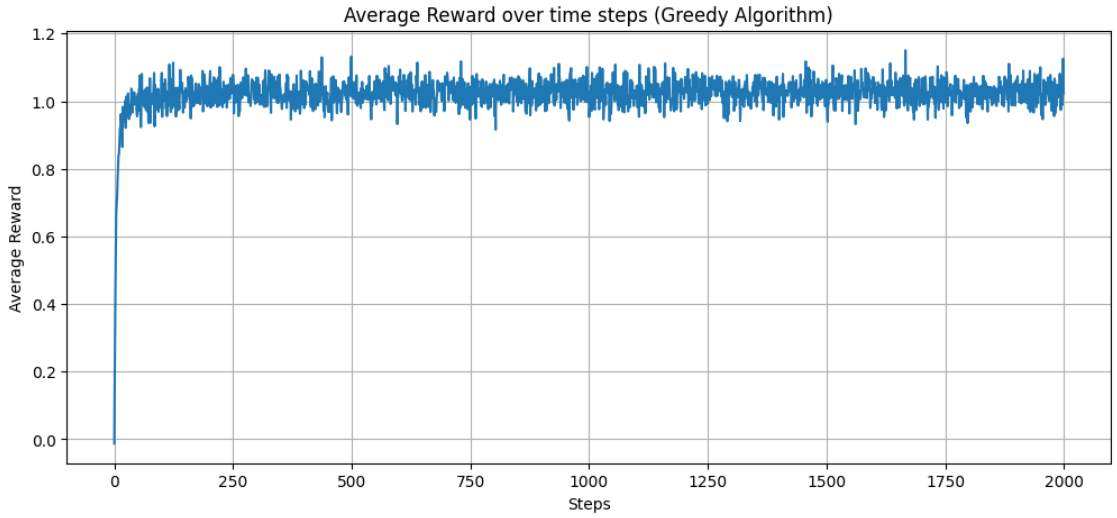

Figure 1: Average reward over time steps (Greedy algorithm)

Figure 1 illustrates the average reward over time for the Greedy algorithm. The average reward increases rapidly during the initial steps, reaching approximately 1.0 within the first 100 steps. After this early rise, the reward stabilizes and remains close to 1.0 throughout the remaining time steps. This indicates that the algorithm is able to quickly exploit actions that provide high rewards based on its initial observations.

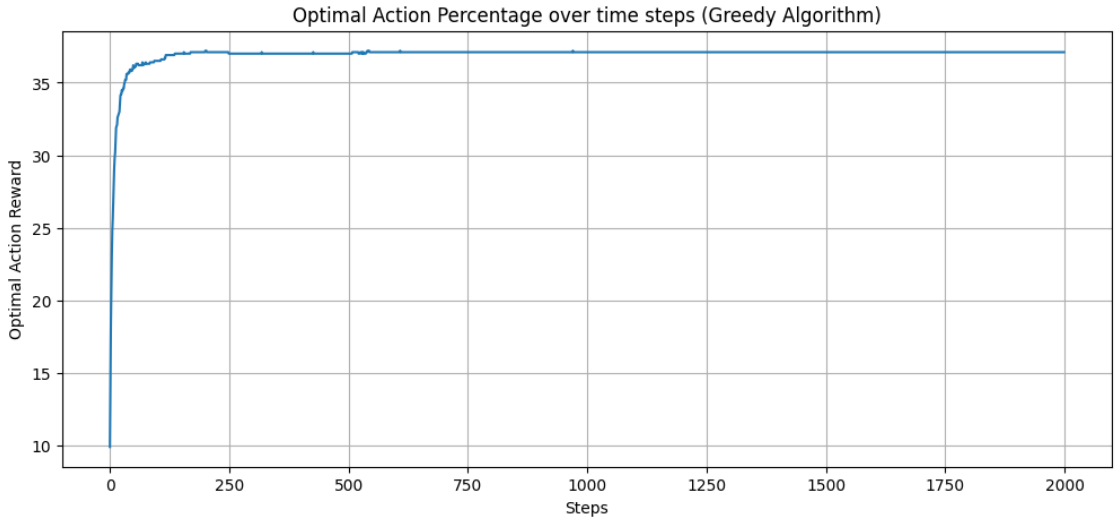

Figure 2: Optimal action percentage over time steps (Greedy algorithm)

Figure 2 shows the percentage of optimal actions selected over time. The algorithm starts by selecting the optimal action around 10% of the time. This percentage gradually increases and stabilizes at approximately 35%. This result highlights a key limitation of the Greedy algorithm: while it is effective at exploiting known high-reward actions, it does not explore enough to consistently find the best possible action.

In summary, the Greedy algorithm performs well in terms of average reward but struggles to consistently identify the optimal action due to a lack of exploration.

1.5.2 Epsilon-Greedy algorithm

Pilot runs to choose epsilon values: A pilot experiment was conducted to systematically evaluate the performance of different epsilon values before committing to full-scale simulations. During this experiment, the parameters were set to arms=10, steps=2000, no of simulations=500 to balance computational efficiency with reliable performance estimation. Moreover, the values considered for the ε mentioned in the first column of the Table 1.

Table 1: Pilot run results with top performers highlighted

Epsilon (ε)	**Final reward**	**Optimal actions (%)**
0.001	1.056	36.6
0.005	1.289	47.3
0.05	**1.418**	**75.7**
0.1	1.397	75.6
0.2	1.367	71.4
0.3	1.126	65.1
0.4	1.022	57.8
0.5	0.776	50.7
0.6	0.595	42.6
0.7	0.422	34.6
0.8	0.327	26.4

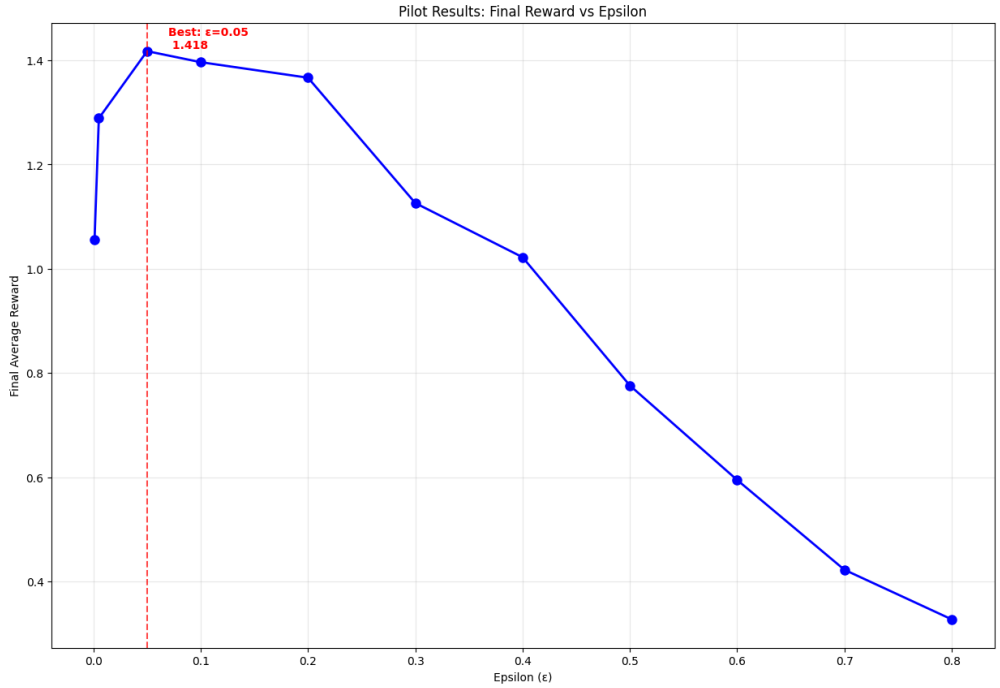

Figure 3: Pilot Results:Final Reward vs Epsilon

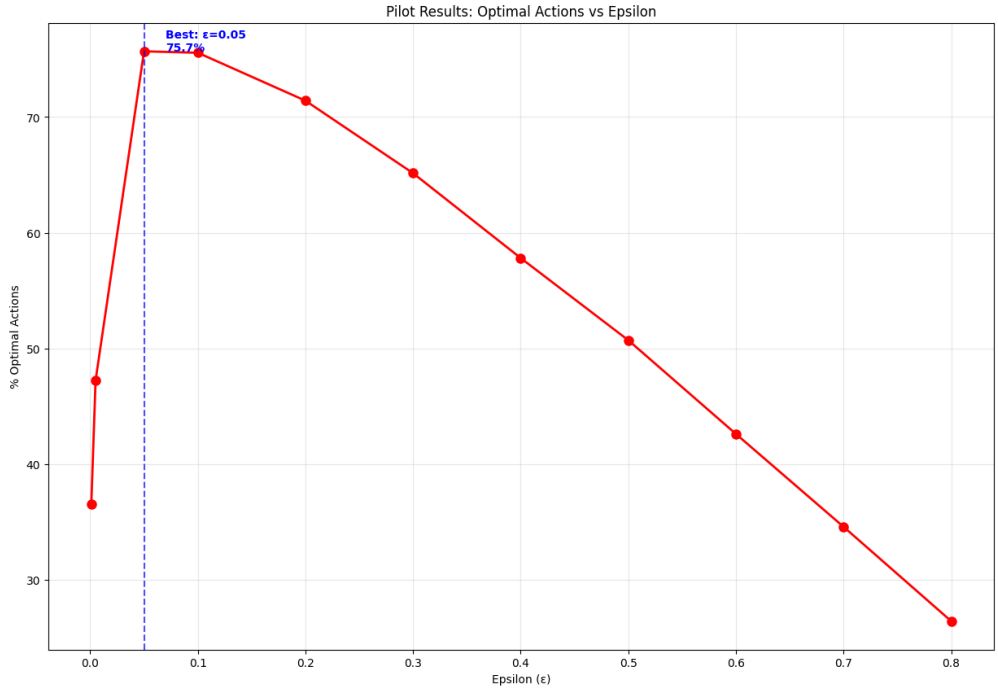

Figure 4: Pilot Results:Optimal Actions vs Epsilon

The pilot experiments revealed significant performance variations across 11 tested epsilon values (Table 1). As shown in Figures 3 and 4, $\varepsilon = 0.05$ achieved the highest final reward and optimal action percentage, highlighted in green in Table 1. Based on these pilot results, the top six epsilon values were identified according to their final reward performance. The top values identified were $\varepsilon = 0.05$ (1.418), $\varepsilon = 0.1$ (1.397), $\varepsilon = 0.2$ (1.367), $\varepsilon = 0.005$ (1.289), $\varepsilon = 0.3$ (1.126), and $\varepsilon = 0.001$ (1.056)

The analysis clearly demonstrates the exploration-exploitation trade-off: very low epsilon values resulted in insufficient exploration, while high values caused excessive random exploration that degraded performance. To ensure robust selection, the evolution of reward curves for each promising epsilon setting was tracked throughout the learning process, allowing identification of epsilon values that provide both good final performance and stable convergence behavior for the comprehensive experimental evaluation.
Based on these top six epsilon values, the epsilon-Greedy algorithm was executed again with extended parameters (10 arms, 2000 steps, 1000 simulations) to conduct a comprehensive performance evaluation and generate detailed learning curves for final comparison.

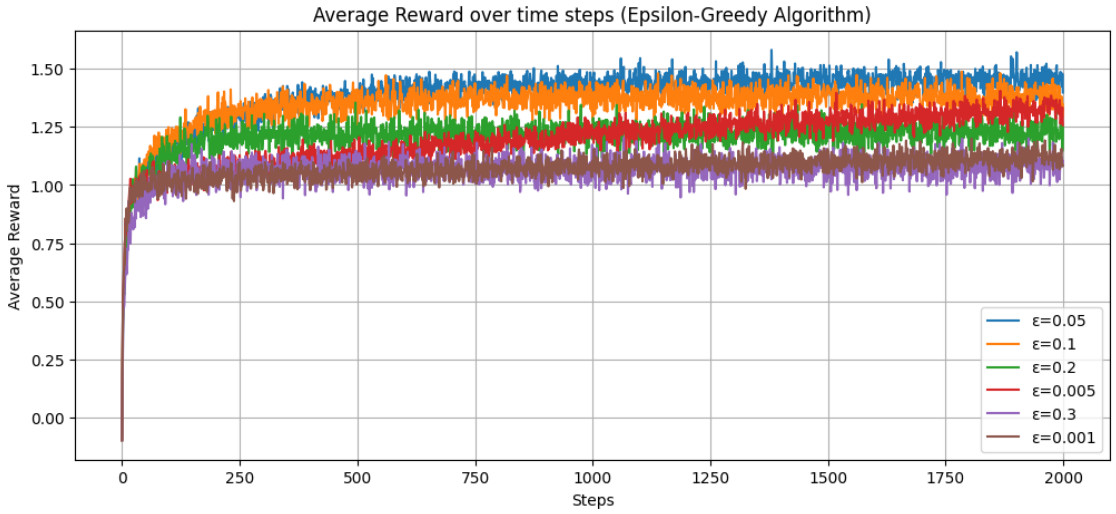

Figure 5: Average Reward over time steps (Epsilon-Greedy Algorithm)

Figure 5 illustrates the average reward evolution over 2000 time steps for the six selected epsilon values in the epsilon-Greedy algorithm. The results demonstrate distinct performance tiers among the different exploration strategies. $\varepsilon = 0.05$ achieved the highest average rewards, converging to approximately 1.5 and stabilizing at this level. $\varepsilon = 0.005$ and $\varepsilon = 0.1$ showed strong performance, reaching steady-state rewards around 1.3-1.35. Very small epsilon values such as $\varepsilon = 0.001$ result in lower average rewards, likely due to insufficient exploration. These settings heavily favor exploitation, making it harder to discover better arms early on. On the other hand, $\varepsilon = 0.3$ introduces more exploration but does not yield better performance, suggesting that too much exploration can hurt the algorithm's ability to consistently exploit the best actions.

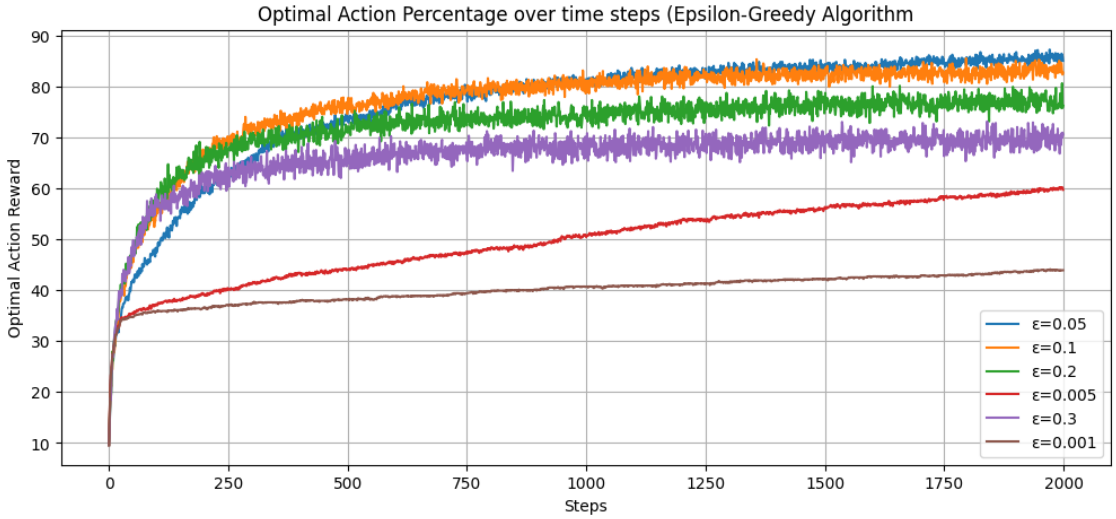

Figure 6: Optimal Action percentage over time steps (Epsilon-Greedy Algorithm)

Figure 6 presents the percentage of optimal actions selected over 2000 time steps for various

epsilon values. The results show that $\varepsilon = 0.1$ and $\varepsilon = 0.05$ achieve the highest percentage of optimal actions, consistently exceeding 80% in the long run. These values balance exploration and exploitation effectively, enabling the algorithm to discover and repeatedly select the optimal arm. The value $\varepsilon = 0.2$ also performs reasonably well but with slightly lower optimal action rates compared to 0.1 and 0.05. In contrast, $\varepsilon = 0.3$ maintains around 65% optimal actions, suggesting that excessive exploration can reduce the algorithm's efficiency in consistently identifying the best arm. The smallest values, $\varepsilon = 0.005$ and $\varepsilon = 0.001$, perform significantly worse, with long-term optimal action percentages remaining below 60%. These low epsilon settings lead to insufficient exploration, often resulting in the algorithm getting stuck with suboptimal arms early and failing to discover better options. Overall, this analysis confirms that moderate epsilon values, particularly in the range of 0.05 to 0.1, provide the most effective trade-off between learning and performance in stationary environments.

So, the Epsilon-Greedy algorithm performs best with $\varepsilon = 0.05$, achieving the highest average reward (1.5) and optimal action percentage (85%). Lower epsilon values (ε) consistently outperformed higher values, demonstrating that conservative to moderate exploration strategies are most effective. Excessive exploration ($\varepsilon = 0.3$) led to degraded performance due to continued random action selection even after optimal actions were identified.

1.5.3 Optimistic Greedy algorithm

To evaluate the effectiveness of the 99.5th percentile optimistic initialization, we tested a range of fixed Q values. Since the 99.5th percentile of a standard normal distribution typically yields values between 2.0 and 5.0 depending on the highest μ_i in each simulation. So, we systematically tested fixed Q values of [2.0, 2.5, 2.75, 3.0, 3.5, 4.0, 4.5, 5.0]. This range allows us to compare the adaptive 99.5th percentile method against static optimistic initialization.

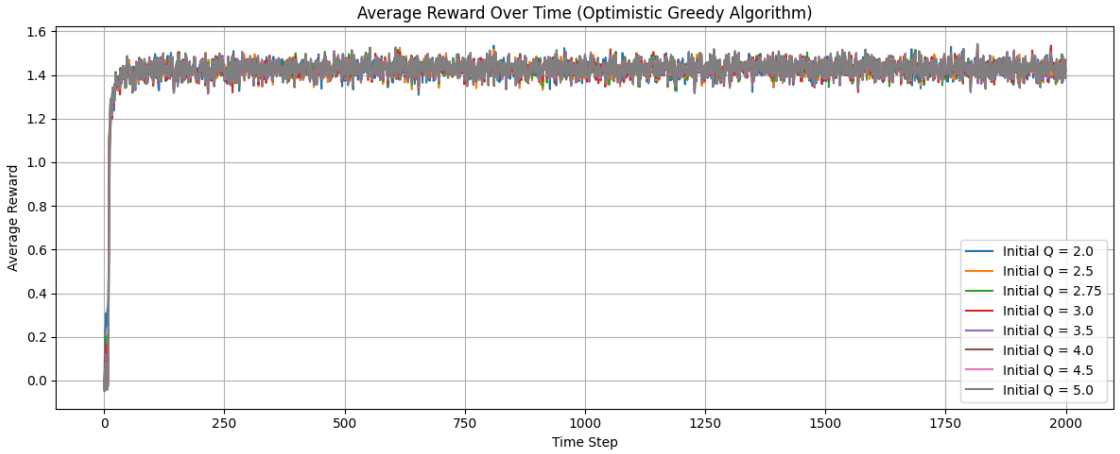

Figure 7: Average reward over time (Optimistic Greedy algorithm)

Figure 7 shows how the average reward changes over time when using different optimistic initial values for Q. All Q values quickly reach high rewards within the first 100 steps. After

the initial learning phase, all lines converge to around 1.4-1.5 average reward. The different Q values (2.0, 2.5, 2.75, 3.0, 3.5, 4.0, 4.5, 5.0) perform almost identically. The optimistic starting value doesn't matter much for final performance. Whether you start with Q=2.0 or Q=5.0, you end up with the same reward after learning.

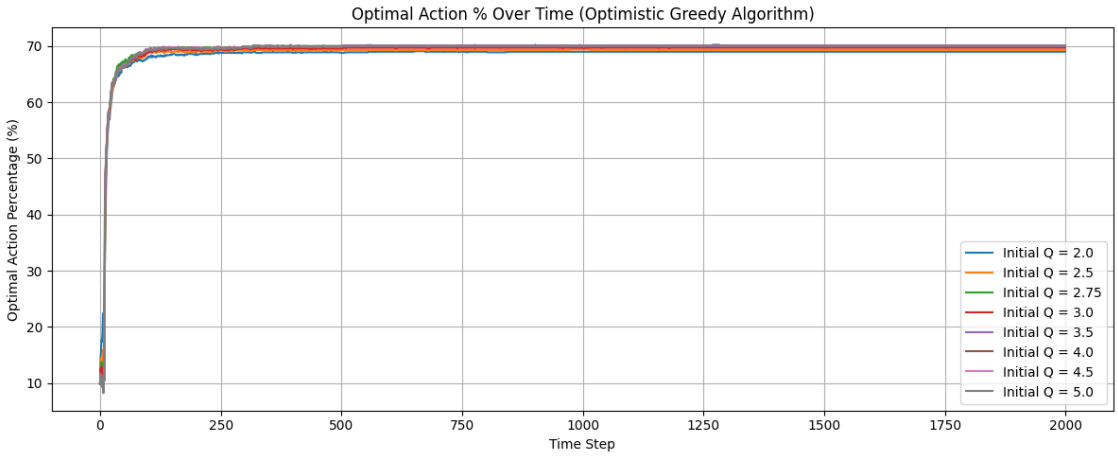

Figure 8: Optimal action percentage over time (Optimistic Greedy algorithm)

Figure 8 shows how often the algorithm chooses the best possible action over time. According to the graph, all Q values rapidly increase from 0% to around 70% within the first 100-200 steps. After step 200, performance levels off at 70-75% for all Q values.
All starting values (2.0 to 5.0) ended up with identical performance. So from this graph it was notices that, higher Q values don't lead to better decision-making in the long run. This confirms that optimistic initialization helps with early exploration, but final performance depends more on the learning process than the starting values.

In addition to Figure 7, 8, and Table 2 presents the final performance metrics for different optimistic Q values after 2000 time steps. The results show remarkably similar performance across all tested values. Moreover, final rewards ranging from 1.386 to 1.427 and optimal action percentages between 69.0% and 70.1%. **This analysis further confirms that Q = 4.0 is technically optimal.** In addition, Figure 9 compares the best-performing fixed Q-value (Q = 4.0) against the adaptive 99.5th percentile baseline approach. The results demonstrate that both methods achieve **identical performance**. So, ultimately, Q = 4.0 considered as optimal value for further analysis.

.

Table 2: Performance analysis of different Q values in Optimistic Greedy algorithm

Q value	Final reward	Optimal action %
2.0	1.386	69.0
2.5	1.427	69.3
2.75	1.421	70.0
3.0	1.417	69.7
3.5	1.421	70.0
4.0	**1.427**	**70.1**
4.5	1.427	70.1
5.0	1.426	70.1

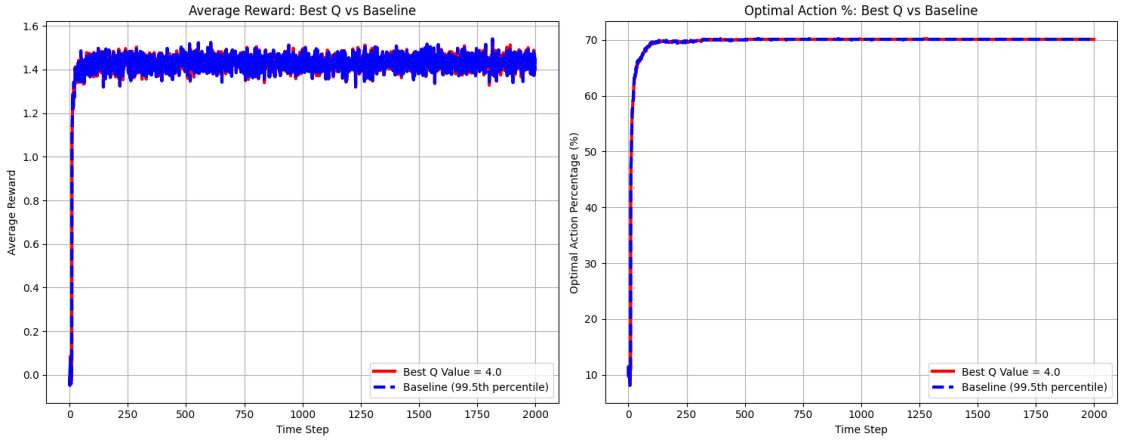

Figure 9: Performance comparison between best fixed Q-value (4.0) and adaptive 99.5th percentile baseline

1.5.4 Gradient Bandit algorithm

To determine an effective learning rate for the Gradient Bandit algorithm, we considered a range of alpha values: [0.01, 0.02, 0.05, 0.1, 0.2, 0.4, 0.5]. This selection covers a comprehensive range from slow learning ($\alpha = 0.01$) to fast adaptation ($\alpha = 0.5$).

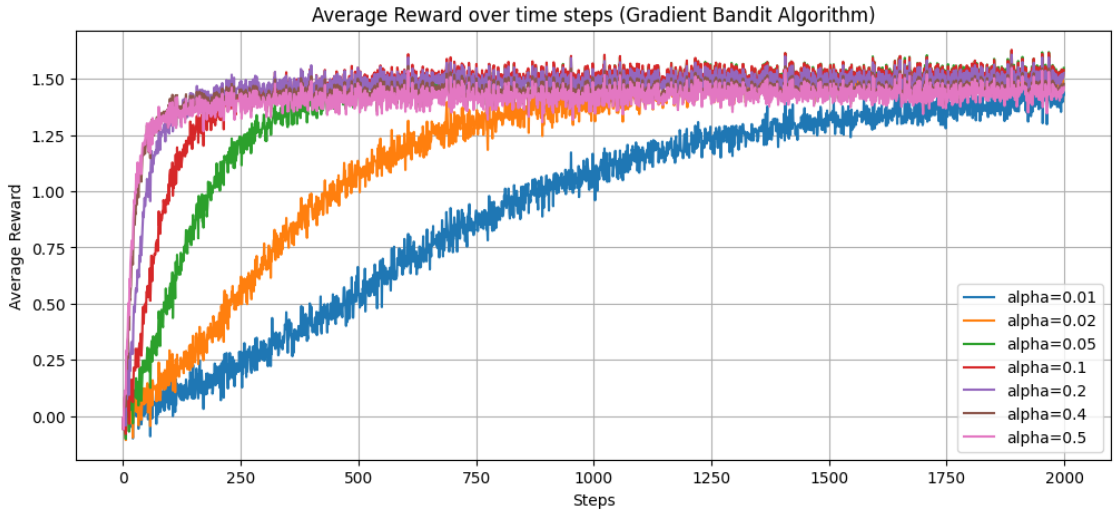

Figure 10: Average reward over time steps (Gradient Bandit algorithm)

Figure 10 shows the average reward performance of the Gradient Bandit algorithm over 2000 time steps using the seven aforementioned different learning rates. Each line represents the learning trajectory for a specific alpha value across multiple simulations. Higher learning rates ($\alpha = 0.4, 0.5$) achieve the fastest convergence, reaching peak rewards of $\sim$1.5 within 200-300 steps, but depicts noticeable fluctuations throughout training.

In addition, moderate rates for α like 0.1, 0.2 also converge to the same high reward levels ($\sim$1.5) but with superior stability and less variance. 0.01, 0.02 α rates demonstrate smooth learning curves but require significantly more time to reach optimal performance. While all learning rates eventually converge to similar final rewards around 1.5, $\alpha = 0.1$ and $\alpha = 0.2$ offer the best balance of fast convergence and consistent performance. So finally, it was clearly noticed that, $\alpha = \mathbf{0.05}$ achieves the smoothest learning curve with minimal variance.

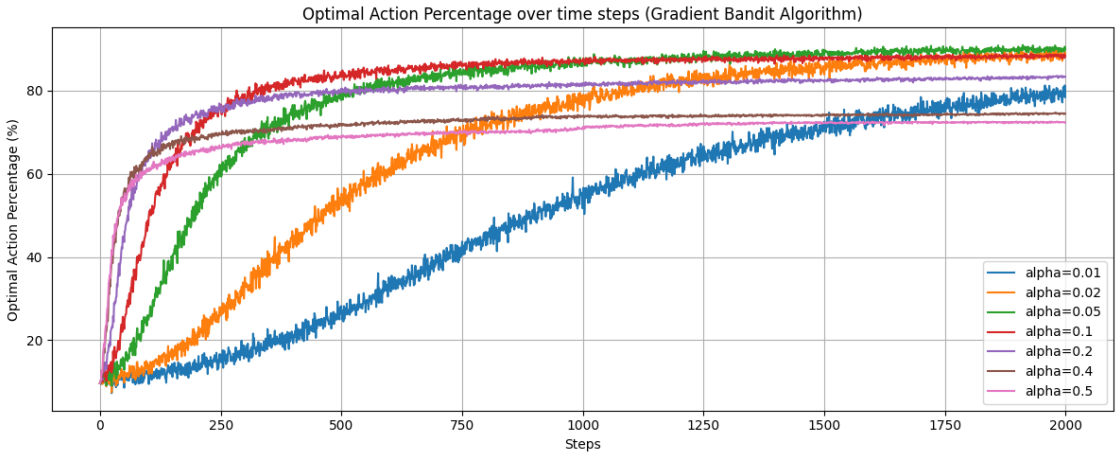

Figure 11: Optimal action percentage over time steps (Gradient Bandit algorithm)

Figure 11 presents the percentage of times the optimal action was selected over 2000 time

steps using the Gradient Bandit algorithm with aforementioned learning rates.

From the results, $\alpha = 0.1$ and $\alpha = 0.05$ achieve the highest optimal action percentage of approximately 88-90% with excellent stability after convergence. In addition, $\alpha = 0.2$ and $\alpha = 0.4$ both achieve solid performance around 82-85%, with $\alpha = 0.2$ showing slightly better stability than $\alpha = 0.4$. Moreover, $\alpha = 0.5$ reaches approximately 80-82%, while demonstrating fast initial convergence but with more fluctuation in the later stages. So, to conclude this $\alpha = 0.1$ and $\alpha = 0.05$ provides the optimal balance of high performance and stability. In addition, $\alpha = \mathbf{0.05}$ serves as an excellent alternative for applications requiring smooth, predictable learning curves.

Table 3 depicts the final performance metrics for the Gradient Bandit algorithm across all the learning rates considered.

Table 3: Gradient Bandit algorithm performance Analysis: final metrics across learning rates

α	**Final reward**	**Optimal Action %**
0.01	1.429	78.4%
0.02	1.523	88.7%
0.05	**1.548**	**89.9%**
0.1	1.539	88.2%
0.2	1.525	83.3%
0.4	1.478	74.4%
0.5	1.463	72.3%

By analyzing the Figure 10, 11 along with Table 3, $\alpha = \mathbf{0.05}$ emerges as the optimal choice for the Gradient Bandit algorithm.

1.6 Comparing all algorithms

To provide a complete performance assessment, we compare all implemented algorithms—Greedy, Epsilon-Greedy, Optimistic Greedy, and Gradient Bandit—using their experimentally determined optimal parameters. Table 4 represents the optimal parameters obtained for the algorithms.

Table 4: Multi-armed bandit algorithms: optimal parameter settings

Algorithm	**Parameter**	**Optimal Value**
Greedy (Non-Optimistic)	Initial Q-values	0.0
Epsilon-Greedy	ε	0.05
Optimistic Greedy	Initial Q-values	4.0
Gradient Bandit	α	0.05

Figure 12 depicts the comparison results obtained for all the four bandit algorithms based on the average reward performance.

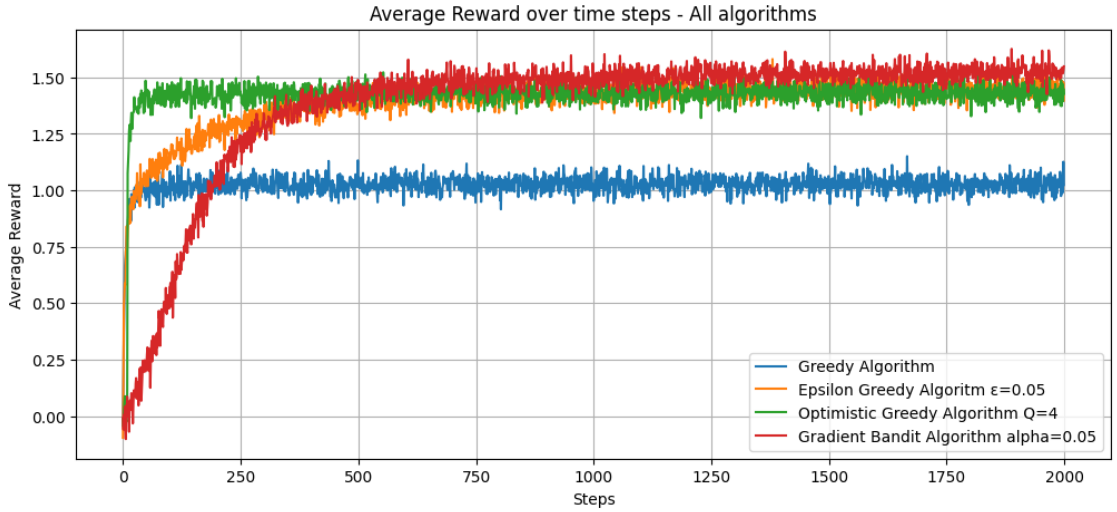

Figure 12: Average reward over time steps - comparing all algorithms

From Figure 12,

- Optimistic Greedy (Q=4) and Gradient Bandit (α=0.05) achieve virtually **identical peak performance**, both converging to approximately **1.5 average reward**. Both algorithms demonstrate **excellent final performance with minimal difference in reward accumulation.**
- Optimistic Greedy: **Fastest convergence**, reaching optimal performance within ∼100-150 steps.
- Gradient Bandit: **Moderate convergence speed**, achieving peak performance around 300-400 steps.
- Epsilon-Greedy (ε=0.05): Slower convergence, requiring ∼500-600 steps to reach plateau.
- Greedy algorithm: Slowest and poorest performance, plateauing around 1.0-1.1 reward.

Optimistic Greedy offers the fastest learning, Gradient Bandit provides comparable final performance with good stability. The pure Greedy algorithm significantly underperforms due to its inability to explore, while Epsilon-Greedy represents a kind of middle solution.

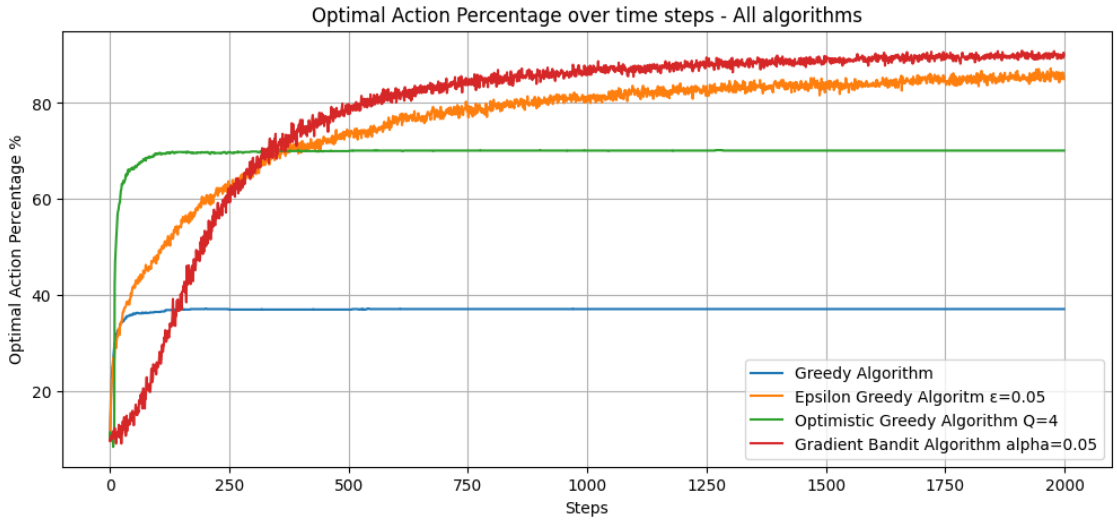

Figure 13: Optimal action percentage over time steps (comparing all the algorithms)

From Figure 13,

- Gradient Bandit (α=0.05) achieves the **highest optimal action percentage**, converging to approximately **88-90%** with excellent stability.
- Epsilon-Greedy (ε=0.05) demonstrates **strong performance**, reaching approximately **85-87%** optimal action selection.
- Optimistic Greedy (Q=4): Fastest initial learning, quickly reaching **70%** within 100 steps but plateauing around **70-72%**.
- Greedy algorithm: Poorest performance, plateauing at approximately **38-40%** optimal action selection.

While Optimistic Greedy offers rapid early exploration, Gradient Bandit provides superior long-term optimal action selection, making it the best choice for maximizing correct decisions over time.

1.7 Conclusion

This comparative study evaluated four multi-armed bandit algorithms—Greedy, Epsilon-Greedy, Optimistic Greedy, and Gradient Bandit—using average reward and optimal action percentage as performance metrics over 2000 time steps.

For clear understanding, Table 5 summarizes the performance ranking of all four algorithms based on their average reward and optimal action percentage metrics.

Table 5: Multi-armed bandit algorithms: performance ranking and final metrics

Rank	Algorithm	Average Reward	Optimal Action %
1	**Gradient Bandit (α=0.05)**	**1.5**	**88-90%**
2	**Optimistic Greedy (Q=4)**	**1.5**	**70-72%**
3	Epsilon-Greedy (ε=0.05)	1.25-1.3	85-87%
4	Greedy algorithm	1.0-1.1	38-40%

The **Gradient Bandit algorithm** (α=0.05) emerges as the superior choice, achieving the highest optimal action percentage ($\sim$88-90%) while maintaining excellent average reward performance ($\sim$1.5). The Optimistic Greedy algorithm (Q=4) demonstrates the fastest convergence and matches the Gradient Bandit's reward performance but falls short in consistent optimal action selection ($\sim$70-72%).

The Epsilon-Greedy algorithm (ε=0.05) provides a reliable middle-ground solution with good performance in both metrics ($\sim$1.25-1.3 reward, $\sim$85-87% optimal actions), while the basic Greedy algorithm consistently underperforms due to its lack of exploration capability.

2 Part 02: Non-stationary bandit problem

Problem setup: 10 arms, 2000 steps, 1000 simulations

Table 6: Optimal algorithm parameters from Part 1 analysis

Algorithm	Parameter	Optimal Value
Greedy algorithm	-	-
Epsilon-Greedy	ε	0.05
Optimistic Greedy	Q	4.0
Gradient Bandit	α	0.05

2.1 Gradual changes

2.1.1 Drift change

$\mu_t = \mu_{t-1} + \varepsilon_t$, with $\varepsilon_t \sim \mathcal{N}(0, 0.01^2)$

Initial testing with 10 seeds (per assignment requirements) produced uninterpretable results with extreme fluctuations and artificial patterns (Figure 14). To ensure interpretability, realistic behaviour, and fair algorithm comparison, we used 1000 unique seeds for all algorithmic implementations, which yielded realistic and analyzable results.

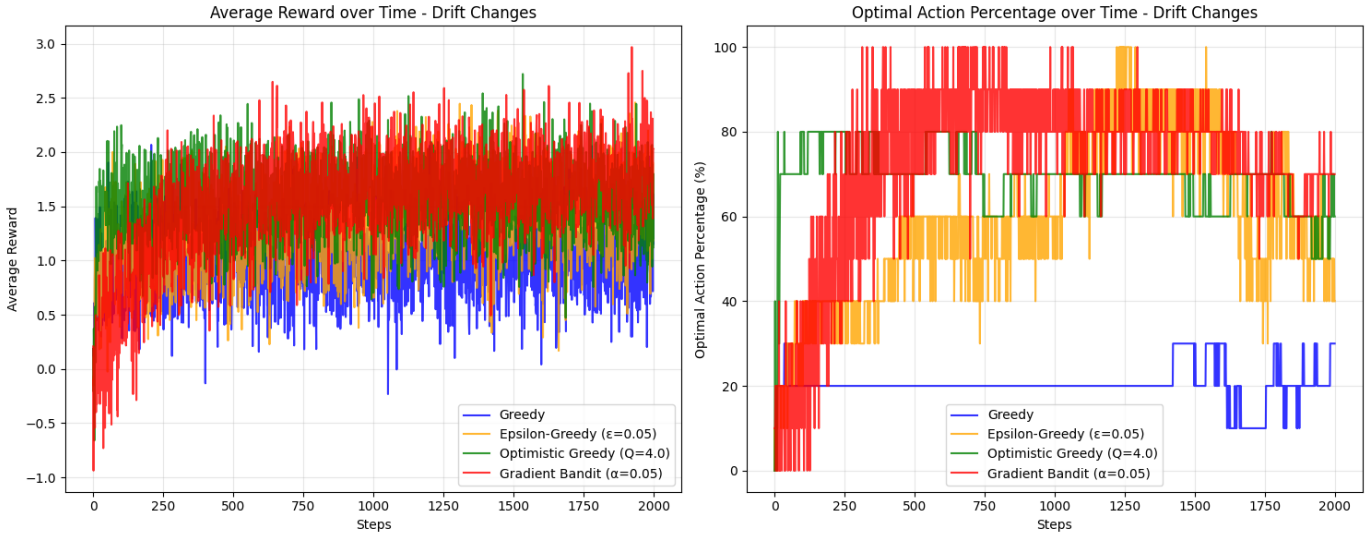

Figure 14: Performance of multi-armed bandit algorithms under drift changes (10 seeds)
Left panel: Average reward over time
Right panel: Optimal action percentage over time

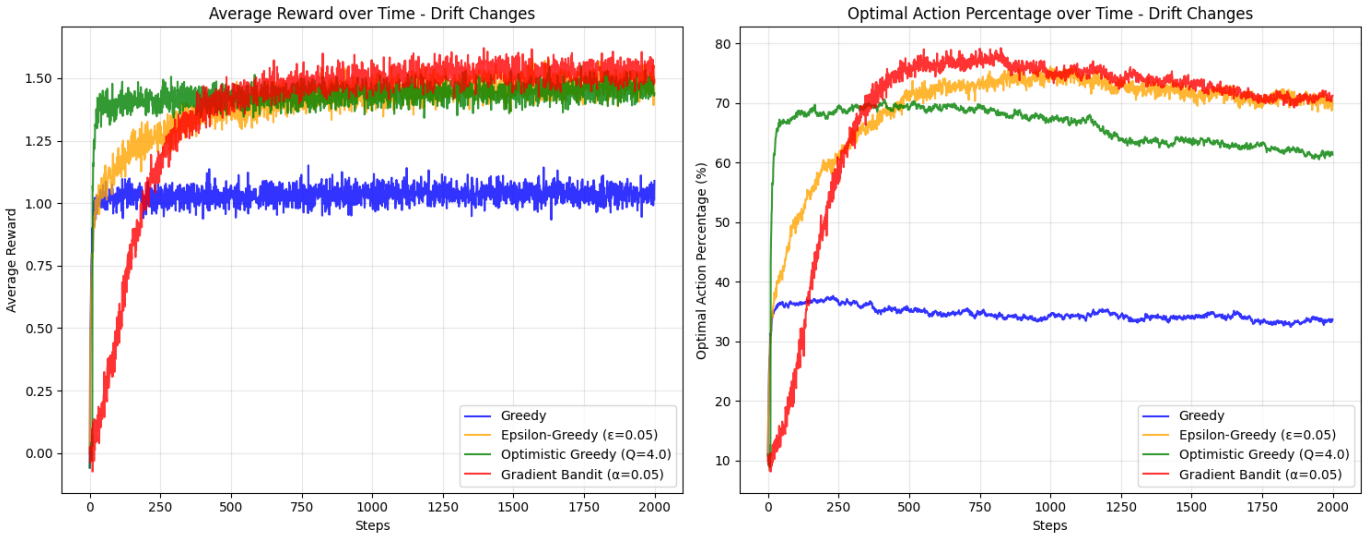

Figure 15: Performance of multi-armed bandit algorithms under drift changes (1000 seeds)
Left panel: Average reward over time
Right panel: Optimal action percentage over time

Table 7: Final performance metrics for multi-armed bandit algorithms under drift changes

Algorithm	**Final Reward**	**Final Optimal%**	**Overall Average**
Greedy	1.089	33.6%	1.031
Epsilon-Greedy	**1.494**	70.0%	1.396
Optimistic Greedy	1.440	61.3%	**1.424**
Gradient Bandit	1.484	**71.2%**	1.374

From Figure 15 and Table 7,

- **Final reward performance:**
  Epsilon-Greedy (1.494) > Gradient Bandit (1.484) > Optimistic Greedy (1.440) > Greedy (1.089)
- **Optimal action selection:**
  Gradient Bandit (71.2%) > Epsilon-Greedy (70.0%) > Optimistic Greedy (61.3%) > Greedy (33.6%)
- **Overall average reward:**
  Optimistic Greedy (1.424) > Epsilon-Greedy (1.396) > Gradient Bandit (1.374) > Greedy (1.031)

Action-value methods vs. Gradient Bandit comparison

Gradient Bandit algorithm

- Achieves the **highest optimal action percentage** (71.2%), demonstrating superior action selection capability
- Shows **rapid initial learning** with smooth convergence to near-optimal performance
- Maintains **consistent performance** throughout the drift period with minimal variance
- Final reward (1.484) places it as the second-best performer

Epsilon-Greedy ($\varepsilon = 0.05$):

- Delivers the **best final reward** (1.494) among all algorithms
- Continuous exploration enables effective adaptation to drifting reward means
- Achieves 70.0% optimal action selection, nearly matching the gradient bandit
- Demonstrates **balanced exploration-exploitation** trade-off

Optimistic Greedy ($Q = 4.0$):

- Achieves the **highest overall average reward** (1.424) across all 2000 steps
- Initial optimistic values provide strong early exploration
- However, shows **declining performance** in later stages as optimism diminishes
- Less effective at sustained adaptation compared to epsilon-greedy

Greedy:

- **Poorest performance** across all metrics due to lack of exploration
- Gets trapped on suboptimal actions as reward means drift
- Unable to discover better alternatives once initial preferences are established

2.1.2 Mean reverting change

$\mu_t = \kappa\mu_{t-1} + \varepsilon_t$, with $\kappa = 0.5$ and $\varepsilon_t \sim \mathcal{N}(0, 0.01^2)$

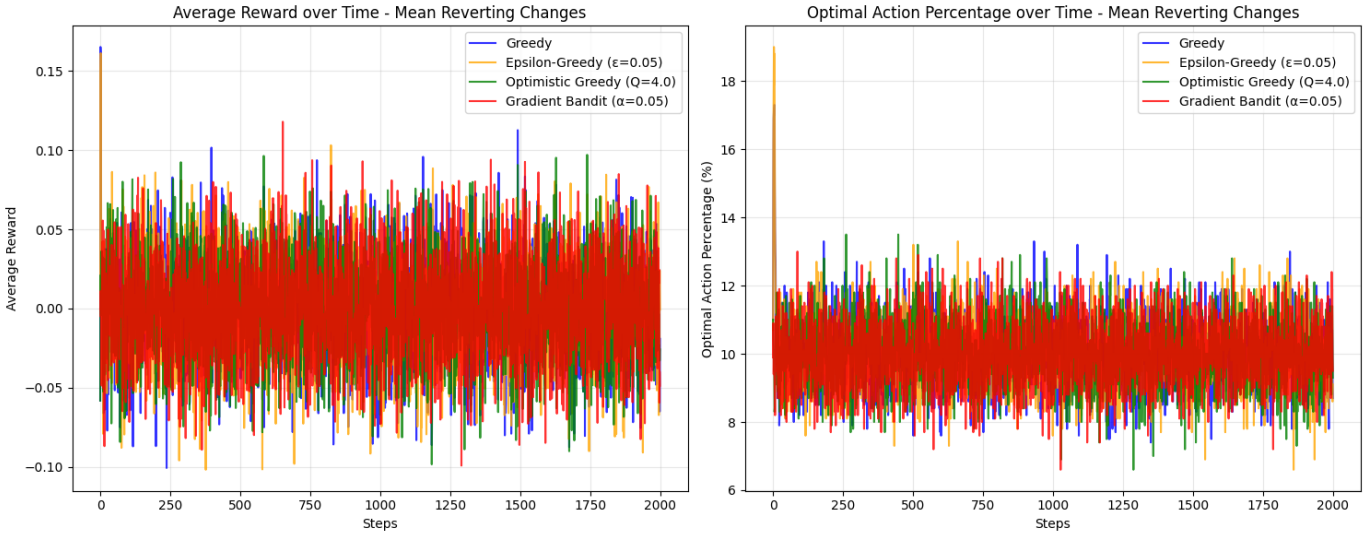

Figure 16: Performance of multi-armed bandit algorithms under mean reverting changes
Left panel: Average reward over time
Right panel: Optimal action percentage over time

Table 8: Final performance metrics for multi-armed bandit algorithms under mean reverting changes

Algorithm	**Final Reward**	**Final Optimal%**	**Overall Avg**
Greedy	−0.019	8.7%	−0.001
Epsilon-Greedy	−0.001	9.1%	0.000
Optimistic Greedy	−0.026	**9.3%**	0.000
Gradient Bandit	−0.049	**9.5%**	**0.001**

From Figure 16 and Table 8,

- **Final reward performance:**
  Epsilon-Greedy (−0.001) > Greedy (−0.019) > Optimistic Greedy (−0.026) > Gradient Bandit (−0.049)
- **Optimal action selection:**
  Gradient Bandit (9.5%) > Optimistic Greedy (9.3%) > Epsilon-Greedy (9.1%) > Greedy (8.7%)
- **Overall average reward:**
  Gradient Bandit (0.001) > Optimistic Greedy (−0.000) > Epsilon-Greedy (−0.000) > Greedy (−0.001)

Action-value methods vs. Gradient Bandit comparison

Gradient Bandit algorithm

- Achieves the **highest optimal action percentage** (9.5%), demonstrating superior action selection capability
- Shows **best overall average reward** (0.001), indicating balanced performance across the entire simulation
- Maintains **relatively stable learning** throughout the mean reverting period
- However, suffers the **worst final reward** (-0.049), suggesting timing misalignment with reversion cycles

Epsilon-Greedy ($\varepsilon = 0.05$):

- Delivers the **best final reward** (-0.001) among all algorithms, closest to optimal performance
- Continuous exploration (5% random actions) provides slight advantages in rapidly changing conditions
- Achieves 9.1% optimal action selection, competitive with other methods
- Demonstrates **most robust adaptation** to oscillating reward patterns

Optimistic Greedy ($Q = 4.0$):

- Achieves **second-highest optimal action percentage** (9.3%) and near-zero overall average (-0.000)
- Initial optimistic values provide early exploration benefits
- However, shows **declining effectiveness** as optimism fades and mean reversion continues
- Performs moderately well but lacks sustained adaptation mechanisms

Greedy:

- Shows **poorest optimal action selection** (8.7%) due to lack of exploration
- Gets trapped on actions that were temporarily optimal before mean reversion
- Final reward (-0.019) demonstrates **limited adaptability** to changing conditions
- Unable to discover new optimal actions as environment oscillates

2.2 Abrupt changes

2.2.1 Abrupt changes as we keep running the algorithms

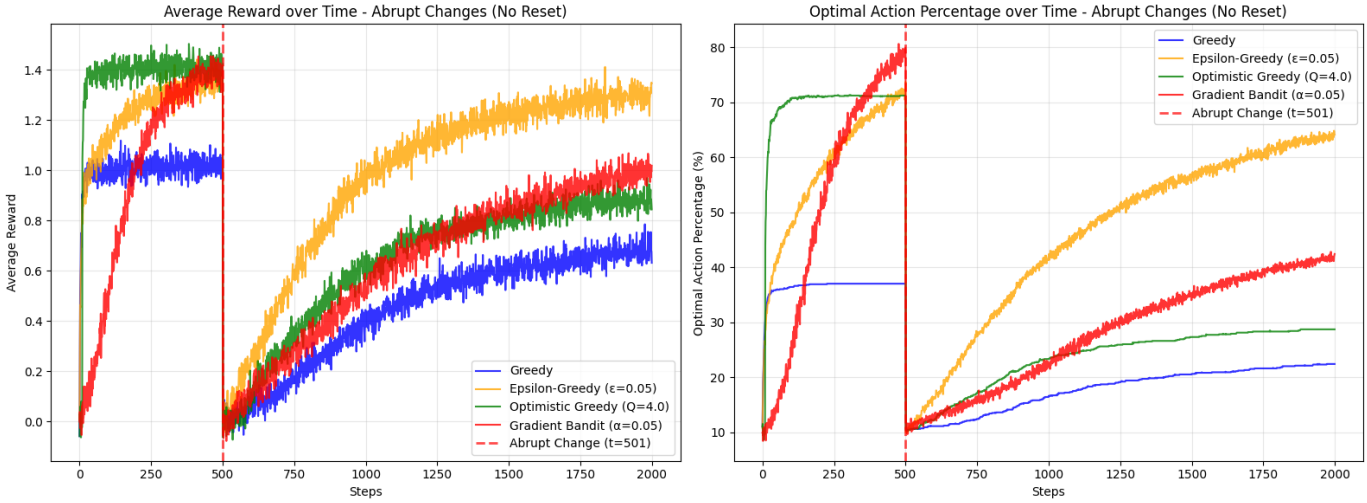

Figure 17: Performance of multi-armed bandit algorithms under abrupt changes
Left panel: Average reward over time
Right panel: Optimal action percentage over time

Table 9: Final performance metrics for multi-armed bandit algorithms under abrupt changes

Algorithm	**Final Reward**	**Final Optimal%**	**Overall Average**
Greedy	0.632	22.4%	0.587
Epsilon-Greedy	**1.346**	**64.1%**	**1.041**
Optimistic Greedy	0.844	28.7%	0.822
Gradient Bandit	1.018	42.4%	0.723

Table 10 depicts the adaptive capacity of different multi-armed bandit algorithms when facing abrupt environmental changes at step 501 during a 2000-step simulation.

Table 10: Performance analysis: adaptation to abrupt environmental changes (no reset)

Algorithm	**Pre-Change Average Reward (Steps 1-500)**	**Post-Change Average Reward (Steps 501-2000)**	**Recovery Steps to Baseline**
Greedy	0.998	0.450	1014
Epsilon-Greedy	1.224	0.981	828
Optimistic Greedy	1.370	0.639	904
Gradient Bandit	0.970	0.640	1084

From Figure 17 and Table 9,

- **Final reward performance:** Epsilon-Greedy (1.346) > Gradient Bandit (1.018) > Optimistic Greedy (0.844) > Greedy (0.632)
- **Optimal action selection:** Epsilon-Greedy (64.1%) > Gradient Bandit (42.4%) > Optimistic Greedy (28.7%) > Greedy (22.4%)
- **Overall average reward:** Epsilon-Greedy (1.041) > Optimistic Greedy (0.822) > Gradient Bandit (0.723) > Greedy (0.587)

Gradient Bandit Algorithm

- Achieves **second-best final reward** (1.018) and optimal action percentage (42.4%)
- Shows **steady recovery** after the abrupt change at step 501, though slower than epsilon-greedy
- Maintains **consistent learning progression** with gradual improvement post-change
- Demonstrates **preference-based adaptation** but lacks the rapid exploration needed for quick recovery

Epsilon-Greedy ($\varepsilon = 0.05$):

- Delivers **outstanding performance** across all metrics, with final reward (1.346) and optimal action rate (64.1%)
- Shows **fastest recovery** after the abrupt change, quickly identifying the new optimal action
- Continuous exploration enables **rapid adaptation** to the new reward structure
- Demonstrates **superior resilience** to environmental discontinuities

Optimistic Greedy ($Q = 4.0$):

- Achieves **third-place performance** with final reward (0.844) and optimal action rate (28.7%)
- Shows **strong pre-change performance** but limited adaptation post-change
- Initial optimistic values lose effectiveness after the abrupt environmental shift
- Demonstrates **insufficient exploration** for discovering new optimal actions

Greedy:

- Shows **poorest adaptation** with final reward (0.632) and optimal action rate (22.4%)
- Gets **permanently trapped** on previously optimal actions after the change
- Complete lack of exploration prevents discovery of new optimal behavior
- Demonstrates **fundamental inability** to handle environmental discontinuities

Table 10 shows how different algorithms adapt to an abrupt environmental change at step 501. Epsilon-Greedy demonstrates superior adaptability with the smallest performance drop (19.9% decline from 1.224 to 0.981) and fastest recovery (828 steps). In addition, other algorithms shows huge discrepancy in performance. In specific, Greedy drops 54.9% and takes 1014 steps to recover, while Optimistic Greedy and Gradient Bandit both fall to similar low performance levels (∼0.64) and require 904-1084 steps for recovery.

2.2.2 Full reset

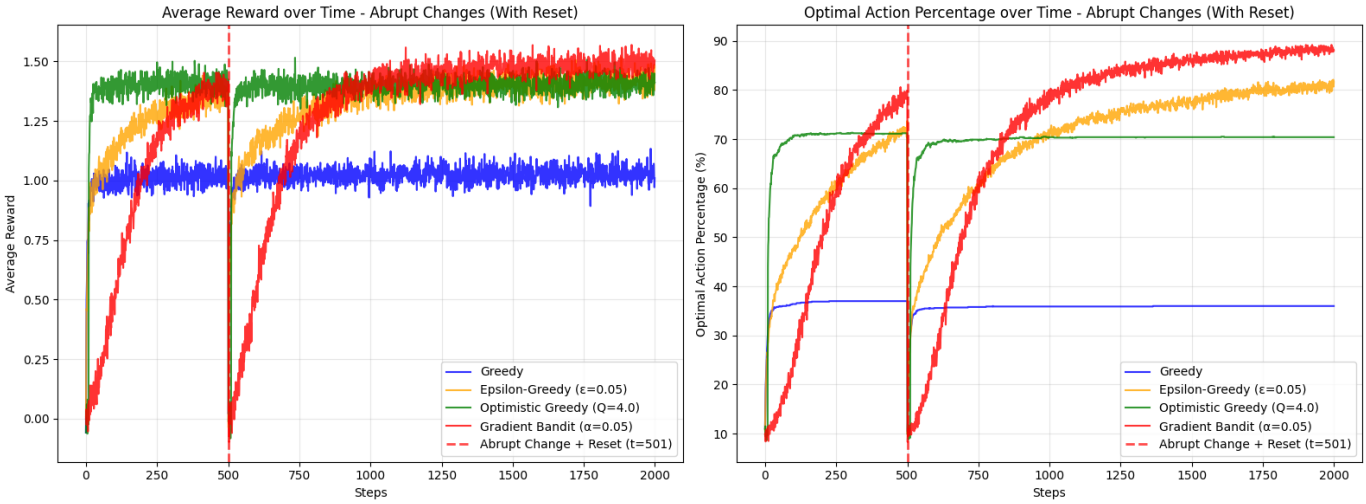

Figure 18: Performance of multi-armed bandit algorithms under abrupt changes (hard reset)
Left panel: Average reward over time
Right panel: Optimal action percentage over time

Table 11: Final performance metrics for multi-armed bandit algorithms under abrupt changes (hard reset)

Algorithm	**Final Reward**	**Final Optimal%**	**Overall Avg**
Greedy	1.010	36.0%	−0.001
Epsilon-Greedy	1.480	81.6%	0.000
Optimistic Greedy	1.411	70.4%	0.000
Gradient Bandit	**1.505**	**88.1%**	**0.001**

Table 12 represents the significant impact of reset mechanisms on algorithm performance when facing abrupt environmental changes in multi-armed bandit problems.

Table 12: Algorithm performance before and after abrupt environmental change (with reset)

Algorithm	**Pre-Change Avg. (Steps 1-500)**	**Post-Change Avg. (Steps 501-2000)**	**Recovery Time (Steps)**
Greedy	0.998	1.019	50
Epsilon-Greedy	1.224	1.344	255
Optimistic Greedy	1.370	1.389	50
Gradient Bandit	0.970	1.309	378

From Figure 18 and Table 11,

- **Final reward performance:** Gradient Bandit (1.505) > Epsilon-Greedy (1.480) > Optimistic Greedy (1.411) > Greedy (1.010)
- **Optimal action selection:** Gradient Bandit (88.1%) > Epsilon-Greedy (81.6%) > Optimistic Greedy (70.4%) > Greedy (36.0%)
- **Overall average reward:** Optimistic Greedy (1.385) > Epsilon-Greedy (1.314) > Gradient Bandit (1.224) > Greedy (1.014)

Gradient Bandit algorithm

- Achieves **best final reward** (1.505) and **highest optimal action percentage** (88.1%), demonstrating superior post-reset learning
- Shows **rapid convergence** after reset, quickly identifying and exploiting the new optimal action
- **Resetting its memory allows rapid adaptation** to the changed conditions
- **Performs excellently when given a clean start** without old learning interfering

Epsilon-Greedy ($\varepsilon = 0.05$):

- Delivers **strong second-place performance** with final reward (1.480) and optimal action rate (81.6%)
- Shows **consistent adaptation** both pre and post-reset, maintaining reliable exploration throughout
- Continuous exploration (5% random actions) enables **smooth transition** after environmental reset
- Demonstrates **robust performance** across different environmental phases

Optimistic Greedy ($Q = 4.0$):

- Achieves **highest overall average reward** (1.385) due to strong performance in both phases
- Shows **dramatic improvement** compared to no-reset scenario, with 70.4% optimal action rate
- Benefits significantly from **renewed optimistic initialization** after reset
- Demonstrates that **optimistic exploration becomes effective** with fresh learning opportunities

Greedy:

- Shows **improved but still limited performance** with final reward (1.010) and optimal action rate (36.0%)
- Benefits from reset but **fundamental exploration limitations persist**
- Unable to fully exploit the fresh start opportunity due to **lack of exploration mechanisms**
- Demonstrates that **reset alone cannot overcome poor exploration strategies**

Table 12 depicts the significance of reset mechanisms in abrupt change scenarios. Unlike the no-reset case where algorithms suffered performance drops, all algorithms now improve or maintain performance after the environmental change at step 501.From the results, Greedy and Optimistic Greedy achieve fast recovery (50 steps each) due to reinitialization benefits, while Gradient Bandit shows the most dramatic improvement ($0.970 \rightarrow 1.309$). At the same time, Epsilon-Greedy maintains the highest absolute performance (1.344) but requires moderate recovery time (255 steps).

Table 13 demonstrates the significant influence of reset capabilities on algorithm performance following abrupt environmental changes. When algorithms can reinitialize after detecting environmental shifts, all methods show substantial performance improvements.

Table 13: Impact of reset mechanism on final reward performance

Algorithm	**No Reset Final**	**Hard Reset Final**	**Reward Improvement**
Greedy	0.632	1.010	+0.378
Epsilon-Greedy	1.346	1.480	+0.134
Optimistic Greedy	0.844	1.411	**+0.567**
Gradient Bandit	1.018	**1.505**	+0.486

2.3 Conclusion

This is the comprehensive analysis of comparison performed on action-value based methods with the gradient bandit algorithm. This was performed across four distinct non-stationary environments over 2000 time steps with 1000 simulations.

Summary of findings

Table 14: Performance comparison across all environments

Environment	**Action vs. Gradient**	**Best Algorithm**	**Final Reward**
Drift	Action-Value	Epsilon-Greedy	1.494
Mean Reverting	Action-Value	Epsilon-Greedy	-0.001
Abrupt (No Reset)	Action-Value	Epsilon-Greedy	1.346
Abrupt (With Reset)	Gradient-Based	Gradient Bandit	1.505

The analysis reveals several critical insights about algorithm performance in non-stationary environments, as summarized in Table 14. Epsilon-Greedy outperformed in 3 out of 4 environments, achieving final rewards of 1.494 in drift, -0.001 in mean reverting, and 1.346 in abrupt no-reset scenarios. In addition, it's continuous exploration provides superior adaptation to most non-stationary conditions. Algorithm performance is fundamentally determined by environmental characteristics, with gradual changes in drift environments allowing both methods to perform well. At the same time, mean-reverting changes cause all algorithms to struggle with only $\sim$9% optimal action rates, and sudden changes favoring exploration-based methods unless reset mechanisms are available. The reset mechanism has a significant impact, as the only scenario where Gradient Bandit outperformed with reset capability. During this environment setup Gradient Bandit achieved a final reward of 1.505 and 88.1% optimal actions compared to 81.6% for Epsilon-Greedy. Furthermore, action-value methods demonstrate superior performance stability by showing more consistent results across different environmental conditions, while Gradient Bandit performance varies dramatically based on environmental characteristics.

3 Code Repository

The full code and README for reproducing results can be found at:
`https://github.com/wbandaramun/reinforcement-learning-assignment-1.git`